COVOX

VOICE MASTER®
junior

USER MANUAL

FOR ATARI 800, 800XL, 65XE, 130XE

SOFTWARE VERSION VMJT 1.0

Includes:

SPEECH RECORDING AND PLAYBACK
SPEECH WORD RECOGNITION
APPLICATION EXAMPLES ON DISK
PROGRAM LISTING

WITH AMPLITUDE EDITOR

LIMITED WARRANTY STATEMENT

COVOX, Inc. guarantees the VOICE MASTER to be free from defective materials and workmanship for a period of one year from the date of purchase. COVOX, Inc. will replace defective parts and make repairs under this warranty when the defect occurs under normal use, provided the unit is returned to the factory via prepaid transportation. The warranty provides that examination of the returned product must disclose a manufacturing defect to be judged by COVOX, Inc. The warranty does not extend to any product which has been subject to misuse, neglect, accident, improper installation, or where the panel legends or other markings have been removed or defaced, and is given in lieu of any other warranty implied or expressed, and will not cover any consequential damages.

Information in this manual and associated software are provided on an "as is" basis. No warranty, either expressed or implied, is made by COVOX, Inc. pertaining to suitability for any specific application or commercial use. It is the purchasers responsibility to make appropriate evaluations for such purposes. COVOX, Inc. disclaims liability for direct, indirect, or incidental damages arising from the use of this product, including but not being limited to interruption of service, loss of business or potential profits, legal actions, or other consequential damages.

Control of environmental factors by means of voice could expose the user to some risk. Word recognition remains an unreliable technology due to uncontrollable variations in the way that normal speech is produced in an uncertain and noisy acoustic environment. Covox, Inc. specifically disclaims liability as stated in the preceding paragraph when applied to word recognition.

PATENTS AND COPYRIGHTS

The software supplied with VOICE MASTER is copyrighted. It may not be copied, reproduced, translated, or reduced to any readable medium or code for other than personal use without prior written permission of COVOX, Inc.

The hardware/software system comprising the COVOX VOICE MASTER is subject to existing patent applications. Unauthorized duplication for commercial purposes or to otherwise avoid payment of appropriate royalties or license fees will be deemed to be a violation of proprietary rights under patent and trademark laws.

The names COVOX, VOICE MASTER, and VOICE HARP, and the COVOX "logo" are registered trademarks and are the property of COVOX, Inc.

RESTRICTIONS ON SOFTWARE USE

Software may generally not be used in programs which are sold or otherwise distributed in violation of copyright laws. There is one exception. Speech that has been produced with Voice Master software may be put into other programs along with playback software, without royalty charges provided (1) software is not for commercial sale, and (2) the source of the speech must be given on the disk jacket, instruction book, and in the disk program itself in sufficient detail to permit a user to acquire a Voice Master. Those wishing to use recognition software and/or edited playback software in programs for sale are advised to contact Covox, Inc. for licensing information.

```
**********************************
* THE VOICE MASTER JR. USER MANUAL *
**********************************
```

by

Kevin C. Gevatosky

TABLE OF CONTENTS

PART 1

Introduction

PART II
Learning To Use The Voice Master

PART III
Recording & Playback

PART IV
Recognition

PART V
Loading And Using The Voice Master Programs

APPENDIX I

APPENDIX II

APPENDIX III

APPENDIX IV
Voice Master Memory Usage Maps

APPENDIX V

APPENDIX VI

APPENDIX VII

IMPORTANT NOTE FOR ATARI 1200XL USERS:

The Voice Master disk will not auto-boot on a 1200XL. The
following procedure will solve this problem.

1. Disable the auto-boot feature by renaming the
"AUTORUN.SYS" file to "AUTORUN".

2. Boot the disk and enter the following:

```
        RUN "D:VM800"
        CLEAR 64
        RUN "D:MENU"
```

The menu will appear on the screen and the Voice Master
will now operate normally.

```
*********************************
              PART I
           Introduction
*********************************
```

This manual is divided into five parts and accompanying appendixes. Part I is concerned with interfacing the Voice Master unit to your Atari and 'booting-up' the disk software. Parts II and III will explain how to use the Voice Master for recording and playback of speech. Recognition of speech will be covered in PART IV and PART V will discuss loading and using the various programs that are on the Voice Master disk. When reading this manual, it is important that you read it from beginning to end without skipping over anything. Each part contains pertinent information concerned with proper operation of the Voice Master unit and software.

Making a copy

Before proceeding any further, a copy of your Voice Master disk should be made and the original kept in a safe place as a back-up only. If something happens to the the copy while using it, then you can always make another from the original. This is a good practice to follow whenever possible and could save you valuable time and money.

Making a copy is easy because we have supplied, for your convenience, ATARI DOS 2.5 on the Voice Master disk. The Voice Master disk has no notch ('write protected') and cannot be written to, but the contents of the disk can still be copied. The software itself is protected under copyright, however you have our permission to make a back-up copy, but only for your own personal use. To make a copy of the Voice Master disk, just follow these simple steps:

1. Turn off your computer and remove any cartridge. If you have an Atari 800 or 1200XL, then insert the BASIC cartridge.

2. Be sure the disk drive is on and set for single density, then insert the Voice Master disk.

3. Turn on the computer and the Voice Master disk will automatically 'boot-up'. When the loading process is finished, the 'VOICEMASTER MENU' will be displayed.

4. Type the number for: 'EXIT TO BASIC' and when the READY prompt appears: type DOS then press RETURN. The DOS 2.5 menu will appear on the screen.

5. Remove the Voice Master disk and insert a blank disk in the drive.

6. Select item: 'J. DUPLICATE DISK' from the menu and press RETURN.
When you see the prompt: 'DUP DISK-SOURCE, DEST DRIVES?'
answer by typing: '1,1' and then press RETURN.

7. Follow the prompts for inserting source and destination disks as required. It will take several swaps of the disks to complete the duplication process.

You should now have a complete copy which will function exactly like the

original. However, this disk is full of programs and there is no room to put your BASIC programs or speech files on it. If you need more space on the same disk, then refer to te section in PART V of the manual on "How to make a customized Voice Master disk."

Setting up the Voice Master

Interfacing the Voice Master Jr. unit with your Atari couldn't be simpler. Just follow the procedure balow and Voice Master Jr. will be 'up and running'.

1. Turn off your computer and remove any cartridges. If you have an Atari 800 or 1200 XL, then insert the BASIC cartridge.

2. Find the joystick port with the number '2' stamped beside it and carefully plug the Voice Master Jr. unit into it.

3. Be sure the disk drive is on and set for single density, then insert the copy of the Voice Master disk.

4. Turn on the computer and the Voice Master disk will automatically 'boot up'. When the loading process is finished, the 'VOICEMASTER MENU' will be displayed.

Adjusting the GAIN control

Getting speech into the computer for recording or word recognition depends on proper operation of a 'voice operated switch' or 'VOX'. A command to record, such as LEARN or RECOG, will cause recording to start when a reasonably intense signal is measured by the VOX and the recording continues until a short period of low intensity signal is measured. If recording is done in a noisy environment, then the VOX will not switch off at the end of speech input and will continue recording noise from the background until the speech buffer is filled to capacity. On the other hand, if input to the microphone is not intense enough to switch on the VOX, then no recording will take place and the buffer will remain empty. Therefore, it is very important that the VOX operates correctly. Proper operation of the VOX is dependent on a combination of GAIN setting, MICROPHONE placement and the LEVEL of voice input.

Adjusting the GAIN control of the Voice Master Jr. is very easy to do with a visual aid. So, select the 'BARGRAPH' option from the 'VOICEMASTER' menu and the program will automatically load into memory. Now, hold the Voice Master Jr. about 3 or 4 inches away from your mouth and continuously make an 'AHH' sound into the microphone. When you do this, you should see a display of vertical bars on the screen. If not, locate the GAIN control knob on the side of the Voice Master Jr. and turn it back and forth through the full range of its motion until the bars appear. The RED light on the Voice Master Jr. should also 'flicker' on and off as the bars move up and down. If no bars appear and the light does not 'flicker' at all, then recheck to make sure that the Voice master unit is firmly plugged into joystick PORT #2 of the computer.

The bar-graph display is a histogram composed of 14 bars which represent frequency response from the microphone. Frequency values are plotted from low on the right side to high on the left side of the screen. Try speaking various sounds or whistles to see how the bar-graph responds to them.

In order to get mazimum sampling efficiency from the Voice Master jr., it is necessary to 'mask' out any background noise that may be present. To do so, turn the GAIN control knob to its maximum setting (clockwise) and then slowly turn the knob back toward minimum setting (counter-clockwise) until the frequency bars on the screen no long respond to background noise but readily respond to your voice input. Different recording environments will require different settings of the gain control. However, with a little practice, you should be able to adjust the gain control for maximum efficiency without the need of the bargraph program by simply observing the level indicator light on the unit.

Microphone Technique

The microphone which is built into the Voice Master Jr. is of the 'electret' type and was carefully selected to provide optimum performance with the Voice Master unit. Although other microphones can be used, do not expect an improvement in speech recording or recognition by switching to another, perhaps more expensive, microphone. However, the distance, position and usage of the microphone is critical, and slight variations in these factors can make great differences when recording speech.

(Note: The electret microphone is supplied with plus 5 volts DC through a resistor of about 4.7K ohms. When using a different microphone or getting signals from a different source (such as tape recorder, radio, etc.), it may be necessary to provide a coupling capacitor to block the voltage supplied to the input phone jack from the Voice Master Jr. A low impedance dynamic microphone may not require a blocking capacitor. But a ceramic microphone substitute may require this.)

Correct positioning of the microphone is essential. First, be sure the element of the microphone is facing toward you. Second, if the microphone is positioned too close to the mouth, breath noise and lip smacking will be recorded and if positioned too far away, important voice sounds will not be detected at all. Try this: Say a word like "hello" into the microphone and watch how the bar-graph responds to the input. Repeat this with the microphone positioned at various distances and angles from the mouth. Did you notice the significant differences an inch or two can make? For most peopole, the microphone should be positioned somewhere between one and a half to three inces from the mouth. But you will have to experiment to determine the exact distance that is right for you. Another factor that will greatly affect speech recording is volume of your voice. Try to speak into the microphone as you would speak to another person who is seated across from you in the same room. It is <u>not</u> <u>necessary</u> to shout, just speak as you normally would. Also, practice to speak consistently and you will get good results with recognition.

Practice your microphone technique with various words and sounds until you feel comfortable with using the microphone. When you want to quit, exit the 'BARGRAPH' program by pressing the SPACEBAR, or any other key, and the 'VOICEMASTER MENU' will be displayed again. Now select the 'DEMONSTRATION' option from the menu. After a few moments the program will load into memory and display a second menu. Choose any option you desire. Just type the corresponding letter and follow the prompts. When you have finished experimenting with the demonstration program, select the: 'TO MAIN VOICEMASTER MENU' option.

```
*************************************
             PART II
    Learning To Use The Voice Master
*************************************
```

The Voice Master unit's controlling software is a sophisticated machine language program which operates via wedged in BASIC commands. This 'wedge' adds new commands to the standard repertoire of Atari BASIC commands. All the added commands were designed to be compatible with Atari BASIC and follow the same general format as the standard commands. The wedge operates in both 'Direct Mode' and 'Program Mode'.

   In Direct Mode, the command is executed immediately after being entered. Just type the command (no line number) with the appropriate parameter and press RETURN. Remember, as with any BASIC command in Direct Mode, you must press RETURN after the command in order for the computer to know that you have entered it.

   In Program Mode, the command is entered within a BASIC line which has a line number, and hence, becomes part of the BASIC program. Multiple commands within a single program line are also supported by the wedge (as they also are in non-wedged Atari BASIC). See APPENDIX V for more details on using the wedge with Atari BASIC.

   There are 19 (20 for Atari 800) added BASIC commands available to you. With these added commands, even the novice who has little or no programming experience, can use the Voice Master to record, playback, and recognize speech.

   Most of the commands require that a parameter, 'n', be specified, whereas with other commands the parameter is optional or unnecessary. A parameter can be entered as a numeric constant (0,45,3 etc.) or given a legal variable name (Z,WORD,X2 etc.). Any variable which is not an Atari BASIC or Voice Master command and begins with an alphabetic character is legal. If a command is entered with an incorrect parameter then an ERROR-3 will result. If the command is within a BASIC program then the ERROR-3 will not occur until the program is executed, whereupon BASIC will STOP execution and display the line number where the ERROR-3 occurred.

Getting familiar

   Now it's time to get introduced to your new Voice Master. We start here with some easy programming in Direct Mode. For now, just follow the given examples. Details about all the available commands will be thoroughly covered further on in this manual.

   If you have not already booted the Voice Master disk, then please do so now according to the directions given in PART I of the manual.

   From the VOICEMASTER MENU, select the 'EXIT TO BASIC' option. When the READY prompt appears, type NEW and press RETURN. Get the microphone ready for speech and type the command LEARN 1. When you are ready to speak, press RETURN, You will notice an inverse plus sign,'+', in the 'status window' at the upper left corner of the screen. This indicates that Voice Master is in the LEARNing

mode and awaiting your input. (If necessary, adjust the GAIN slightly to stabilize the inverse plus sign.) Pause the LEARNing mode at this point by pressing the UP-ARROW key without holding down the CONTROL key. Actual speech recording will now be suspended and an **inverse up-arrow,'↑'**, will appear in the status window. This feature comes in handy when you are recording in an environment with occasional loud background noise. When you are ready to resume recording your speech, just press any key and the inverse plus sign will return to the window. Now speak into the microphone and notice how different characters 'flicker' through the window as you speak. Each 'flick' represents an amplitude byte of speech data being sampled. When you stop speaking the window will 'vanish' from the screen and the READY prompt will reappear. If the READY prompt reappears before you have finished speaking, then you have either paused too long between words and the program assumed you were finished speaking, or you did not speak with enough volume in your voice. If you here a 'beep' when recording, then you have exceeded the capacity of the speech input buffer (capacity is about 4 seconds for Atari 800's and about 7 sec. for XL's and XE's). So, use words or phrases that are short enough to fit within the capacity of the buffer.

Type **SPEAK 1** and press RETURN. The computer will now play back the recorded word. Be sure the volume on your T.V. or monitor is set high enough to hear. Now, press the SPACEBAR during playback of the word and the word will re-play from the start. If the SPACEBAR is rapidly and repeatedly 'tapped' during playback, it will make an interesting reverberatory sound effect. If any key besides the SPACEBAR is pressed during playback, then the routine will stop and the READY prompt will return. Because word #1 remains in memory, you can SPEAK 1 over and over again. Similarly, you can re-record word #1 by entering the command LEARN 1 again, using either longer or shorter utterances.

The loudness of the recorded word can be varied by using the **VOLUME** command. First type VOLUME 7 and press RETURN. Then type SPEAK 1 and press RETURN. Notice the reduced loudness? Now restore the original volume by entering VOLUME 15.

SPEED is a command that produces an interesting sound effect. First, record a word or phrase with LEARN 1, then change the playback speed by entering SPEED 1. Now enter SPEAK 1 and notice the slowed playback. Try entering SPEED 3 and then SPEAK 1. Finally, restore the normal default value by entering SPEED 2.

Saving a speech file is as simple as saving an Atari BASIC program. Use LEARN 1 to record a word or phrase to be saved, then SPEAK it back. Remove the Voice Master disk from the drive and insert a blank formatted disk to be used for storage of your speech files. Then type **SSAVE"D:SPEECH"** and press RETURN. Your recorded speech is now safely stored on disk. Type CLEAR and press RETURN to erase all speech stored in memory and then enter SPEAK 1. You will here a 'beep' signifying that there is now no recorded speech in memory. To load the speech file back into memory, just type **SLOAD"D:SPEECH"** and press RETURN. Enter SPEAK 1 and you will hear that the word you recorded and SSAVEd is back in memory where you can re-record it, if you like. The filename "D:SPEECH" was used above, but this could have been any filename of your choice, such as "D:MYWORD".

As previously stated, all Direct Mode commands can also be entered in the Program Mode as numbered program statements. This means that BASIC programs can be written which include speech recording and playback! And the variety of ways

in which you can incorporate Voice Master commands in your BASIC programs is limited only by your imagination.

Using the CLEAR command

Before starting in on some actual examples in Program Mode, speech and program memory should be cleared. In order to erase your vocabulary from memory, simply type CLEAR. On the Atari 800 and XL models, speech is stored in user-ram above RAMTOP (location 106). The area below RAMTOP is available for the graphics display list, screen data, and your BASIC program. The amount of ram available for your BASIC program can be increased or decreased, according to your needs, by entering the 'CLEAR n' command. The variable 'n' can be any value between 48 and 118 for the Atari 800 or 48 and 156 for the 800XL. This number specifies the 'page' in memory where speech data is to be stored. To maintain the graphics display, memory should be reserved in 4K increments. So, use values of 'n' that are multiples of 16, for example: CLEAR 64, CLEAR 96 etc. If the CLEAR statement without a value for 'n' is used, then the default value will be set at 48 for the Atari 800 and 64 for the 800XL; or whatever value was last assigned to 'n' with the 'CLEAR n' command. No value for 'n' is needed, nor permitted, with the CLEAR command for the ATARI 130XE. Speech data on the 130XE model is stored in the extra 64K bank of memory and all of the ram that is normally available to BASIC is still available with the Voice Master software loaded and running.

The CLEAR command erases speech data stored in memory, but does not erase or affect a BASIC program. Likewise, the NEW command erases BASIC program memory and variables, but does not erase recorded speech or affect the Voice Master program in any way.

Whenever you start a new recording session you may wish to CLEAR. This will ensure that leftover speech data from a previous recording session is erased. You may also want to NEW, so as to remove any statements left over from some previous BASIC program and also reset BASIC's pointers and variables.

Voice Master commands in Program Mode

The following program allows you to enter 12 words in Program Mode. Prompts to the screen are included. (If you can't think of some words, just count!)

```
10 FOR WORD=1 TO 12
20 PRINT "SAY WORD NUMBER ";WORD
30 LEARN WORD
40 NEXT WORD
```

You can hear each word after recording it if you insert the statement:

```
35 SPEAK WORD
```

If you don't like any of the words learned, then you can replace them in Direct Mode by typing LEARN 2, LEARN 5, etc. This will not affect any of the other recorded words.

Now erase the BASIC program with NEW, but do not CLEAR the speech. The 12 words still remain in memory. You can now use the same words in another program, in any order. Here is an example using a data statement to assign values to the

SPEAK command.

```
10 RESTORE
20 READ WORD
30 SPEAK WORD
40 GOTO 20
50 DATA 5,4,10,1
```

Now RUN the program. An ERROR-6 (out of data error) will occur when the program
has finished running.
If words SPEAK back too close together in time, then insert this simple delay
loop:

```
25 FOR T=1 TO 100:NEXT T
```

Or use the **PAUSE** command, which will be described in PART III of the manual.
But, for the meantime, try PAUSE 3 or PAUSE 6 (e.g. 25 PAUSE 6). Other commands
like SPEED 2, VOLUME 5, and SLOAD can also be included and the Voice Master
BASIC program saved in the same manner as a standard BASIC program.

```
****************************
          PART III
    Recording & Playback
****************************
```

This part of the manual describes those commands which are used solely for speech reproduction. If you followed the examples given in the "Getting familiar" section, then you are already acquainted with most of them.

Any sounds within the sensitivity range of the Voice Master unit can be recorded. This includes short monosyllabic utterances, words, phrases, whispers, bells, whistles, humming, clapping, hammering, sawing and many more. When speech (or any sound) is recorded with the LEARN command, it is converted into digital data ('digitized') and simultaneously stored in an area of memory designated as the 'speech input buffer'. After the speech has been input, the digitized data is then moved, byte by byte, into another part of memory reserved for the storage of speech data. The recorded speech is then ready to be played back at any time with the SPEAK command.

The LEARN and SPEAK commands

Words are numbered from 0 to 63 and up to 64 different words or phrases can be in memory at one time. A word number must be specified when using the LEARN or SPEAK commands. 'LEARN n' and 'SPEAK n' are the correct format, where 'n' can be assigned any constant or variable that has a value within the range indicated above. Each word number, 'n', functions as an index to that particular word or phrase which was recorded and stored in memory. After a word has been LEARNed, it can be accessed again and again using the same word number. If you don't like a particular word, then simply re-record it using 'LEARN n'. Re-LEARNing a word or phrase replaces the original one with the new one. This usually requires that other speech data be moved up or down in memory to accommodate the new word and a varied amount time is required to do so. An inverse 'W' will appear in the 'status window' when you finish speaking a word. This indicates that the computer is 'Working' to move speech data from the input buffer to the main speech storage area. The amount of time (usually a fraction of a second) required to move speech data from the buffer to the storage area depends on the amount of speech data already present in the storage area and the model of Atari being used. The 130XE model can store over 60 seconds of speech data in the extra 64K bank of memory and must also do bank switching, so it takes a little 'bit' longer.

The maximum length of a word that can be LEARNed is limited by the capacity of the speech input buffer. If you attempt to LEARN a word or phrase that is too long, or SPEAK one that has not been previously recorded, then you will hear a telltale **'beep'**. You will also hear the beep if you have used up all of the memory allocated for speech storage. There are times when it is difficult to discern what caused the beep, especially when it occurs within a BASIC program. Therefore, the Voice Master uses memory LOCATION 209 as an error number register and for recognition purposes. See APPENDIX III for a complete list of error codes. When you here a beep, just type in direct mode PRINT PEEK(209) and press RETURN to see what error occurred. Error checking may also be done from within a BASIC program. For example, the following program will print an error if the speech buffer overflows:

```
10 LEARN 0
20 IF PEEK(209)=253 THEN 40
30 SPEAK 0 :PRINT "Successful LEARN."
40 PRINT "ERROR! Buffer overflow!"
50 POKE 209,128 :REM Acknowledge error.
60 END :REM End program.
```

Keyboard control

When the learn command is entered, either in Direct Mode (from the keyboard) or from Program Mode (through BASIC program control), a machine language recording routine is activated which takes control of the computer and awaits input from the microphone. If you want to pause the routine at this point, then press the UP-ARROW key '^' and all input to the microphone will be ignored (this produces the same effect as an 'ON/OFF' microphone switch). Press the UP-ARROW again, or any other key, and the recording process will be re-initiated. If you change your mind and decide not to record a word at all, then you can exit the record routine entirely by pressing the BREAK key. The BREAK key may also be used to terminate the SPEAK command before playback has finished. In either case, pressing the BREAK key will immediately, and always, return control to the keyboard (unless the BREAK key was previously disabled by your BASIC program). When a LEARN or SPEAK command is entered in Direct Mode, then pressing any key (with the exception of BREAK, UP-ARROW and SPACEBAR (during playback)) will cause the record or playback routine to terminate, print the ATASCII value of the key-press on the screen and return control to the keyboard. However, if the LEARN or SPEAK command was initiated in Program Mode, then control will return to the program and the key-press will not be printed to the screen. However, LOCATION 209 can be PEEKed by the BASIC program to determine if a key was pressed. The following program demonstrates how to detect a key press during playback and subsequently print the value of the key to the screen. This routine also works with the LEARN command. First enter LEARN0 to record a word, then RUN this program:

```
10 OPEN #1,4,0,"K:" :REM Open keyboard.
20 SPEAK 0
30 IF PEEK(209)<>251 THEN 20:REM Keep SPEAKing?.
40 GET #1,A :REM Get ATASCII value of key.
50 PRINT CHR$(A) :REM Print it to the screen.
60 POKE 209,128 :REM Akcknowledge error
70 GOTO 20 :REM and continue.
```

The TIME-OUT feature

After approximately two minutes, without input into the microphone, the record routine will automatically be exited and control will return to BASIC. This is the result of a 'time-out' feature which uses a value,'n/2 seconds', in LOCATION 207 to determine how long the routine is to wait for an input. Any number from 1 to 255 or 0 can be POKEd into this location to change the 'waiting' time. For example, POKE 207,20 will cause the record routine to time-out after only 10 seconds. A zero is the default value and is equivalent to a time-out duration of 256/2=128 seconds. Here is a sample program which demonstrates how the time-out feature could be used:

```
10 POKE 207,10 :REM Set time-out for 5 sec.
20 LEARN 0 :REM Record something.
30 IF PEEK(209)=250 THEN 60 :Time-out yet?
40 SPEAK 0 :If not then SPEAK something.
50 PRINT"Thank you.":GOTO 20
60 PRINT"Please say something."
70 POKE 209,128 :REM Acknowledge error
80 GOTO 20 :REM and continue.
```

The CALIB command

The CALIB command can be used in either Direct or Program Mode to check the calibration setting of the Voice Master. When CALIB is entered, the record routine is initiated and an inverse question mark,'?', appears in the status window'. Although the record routine will input sound from the microphone, no sound data will be stored in memory. Once initiated, the CALIB command causes the record routine to continue inputting data until either a key is pressed or time-out occurs. LOAD and LIST the "MENU" program for an example of using CALIB within a BASIC program and for instructions on calibrating.

The VOLUME command

The loudness of a word or phrase can range from 1 to 15. The default value is set at 15, which is the maximum VOLUME setting. Use 'VOLUME n' to adjust the loudness of words or phrases that are played back with the SPEAK command. The VOLUME command can be used to produce some interesting sound effects, such as echoing a word or phrase during playback.

The PAUSE command

This command acts exactly like a timing loop in BASIC (e.g. 10 FOR T=1 TO 10:NEXT T). However, 'PAUSE n' is easier to use. The PAUSE command produces a fixed timing delay of 1/10 second. The 'n' parameter specifies the number of tenth second increments, with a range of 1 to 255. For example, PAUSE 20 will give a 2 second delay. For timing delays of less than 1/10 second, use a BASIC timing loop or a 'dummy' time wasting statement.

The RATE command

When speech is recorded, the Voice Master hardware samples both the frequency and amplitude of the incoming speech and converts this information into binary bits of data. This process is known as digitization. After the speech has been digitized, it is stored in the computer's memory as a series of data bits. A single bit of data in memory represents a moment in time at which a sound pulse from the incoming speech was detected. If a pulse was detected, the bit will have a value of one, otherwise it will be zero. The number of times that the incoming speech is sampled, per given period of time, is called the 'sampling rate'. The sampling rate at which the Voice Master samples speech, in conjunction with the Atari, can range from under 3,600 bits per second to 15,600 bits per second. Higher sampling rates reproduce speech with greater precision and somewhat better reproduction quality. But, this is at the cost of consuming greater amounts of memory to store the recorded data. Lower sampling rates require less memory to store the speech data, but the reproduction quality is not as good as with the higher rates.

There are three sampling rates that can be used during the recording of speech. The 'RATE n' command can be used to select low (1), normal (2) or high (3) sampling rates. RATE 2 is the normal default setting and corresponds to an actual sampling rate of about 7,800 bits per second. This rate is adequate for most purposes. Use the RATE 1 command when memory space is critical. With RATE 1 you can store up to twice as much speech in memory as is possible with RATE 2. The 'RATE n' command sets both recording and playback speed to the same value. If RATE 3 is used to LEARN a word or phrase, then SPEED 3 will be used to play it back. If a word or phrase is first LEARNed using RATE 2 and then another LEARNed at RATE 3, SPEED 3 will be selected by the RATE command as the default setting.

Using RATE 3 will give the best quality speech attainable with your Atari. However, this sampling rate is so fast that the 6502 microprocessor should record and playback speech without interference from the ANTIC chip which 'steals' cycles from the 6502 to update the screen. If you want to prevent ANTIC from stealing cycles, use the SCREEN command to turn off the screen during recording and playback of speech. (See below.)

Another feature of the Atari (and most other computers), that affects recording and playback quality, is the 'Vertical Blank Interrupt' or **VBI**. The VBI occurs every 1/60 of a second and if it were not disabled then speech input and output would sound 'choppy' at high rates. So, the record and playback routines automatically disable the VBI before inputting or outputting speech. However, disabling the VBI also disables the Atari's 'software clock' and, in most cases, Player/Missile graphics. Therefore, a special 'flag' at LOCATION 1663 ($67F) can be POKEd with a non-zero number in order to enable the VBI during playback only. Just POKE again with a zero to re-enable VBI's. For an example on using this location, see the "CLOCK" program.

For more information on ANTIC and the VBI, see: **"De Re Atari"** by Chris Crawford et al., Atari Program Exchange (Calif.:1982).

The SPEED command

The command 'SPEED n', varies the playback rate of the recorded word or phrase, but has no effect on the recording rate. This allows you to SPEAK back words at rates slower or faster than the rate at which they were recorded. Five speed settings are available, ranging from 0 to 4, with 0 being the slowest and 4 the fastest. Playback SPEEDs 1,2 and 3 correspond to recording RATEs 1,2 and 3 and a word or phrase that is recorded at RATE 1 will sound normal when played back at SPEED 1. The default setting for the SPEED command is '2'.

The SCREEN command

This command gives you the option of turning off the screen while LEARNing or SPEAKing. The format is 'SCREEN n', where 'n' ranges from 0 to 3. Use SCREEN 1 to turn off the screen during recording, SCREEN 2 to turn it off during playback, and SCREEN 3 to turn it off during word recognition (see PART IV). Once the screen has been turned off with the SCREEN command, it will remain off until restored to normal with a SCREEN 0. Entering SCREEN 0, restores all screens to normal (on). As stated above, this command should be used with the the high sampling rate in order to get the best quality of speech reproduction.

The PORT command

The Voice Master unit can operate from either joystick port #1 or joystick port #2 by entering PORT 1 or PORT 2 as appropriate. This feature allows the Voice Master unit to be used with other peripherals, which might only connect with port #2.

The FREE command

There is not nearly as much memory available for speech data storage on the Atari 800 as there is with the 800XL and 130XE models. For this reason, a special command has been added exclusively for the Atari 800.

The FREE command allocates about 4K of additional memory space as an extension to the speech data storage area. Because this 4K of memory is normally used for storing recognition templates (more about this later), no speech recognition can be done after entering the FREE command unless system RESET is pressed first. There are no parameters needed for this command, just type FREE and press RETURN.

The SSAVE and SLOAD commands

A complete vocabulary can be saved on disk or tape in either Direct or Program Mode. When saved to disk, each speech vocabulary is saved with the particular filename that you assign to it. Thus, there is no theoretical limit to the number of vocabularies that can be saved to disk (or tape) and then loaded back for incorporation of speech within your BASIC program(s). The command to save a speech file to disk is, SSAVE"D:FILENAME" and for loading use, SLOAD"D:FILENAME". The filename can be any name which is legal for the particular DOS that you are using. If using a cassette to store and load speech data, then use SSAVE"C:" and SLOAD"C:".

When speech data is SSAVEd, all pointers associated with the data are also saved. This means that when the speech is SLOADed back into memory, it will load back to the same area of memory that it was SSAVEd from and RAMTOP will be set accordingly. Also, all values for VOLUME, SCREEN, SPEED etc., will be restored to what they were when the speech file was SSAVEd. SSAVE and SLOAD are used for storing and retrieving speech data only! The standard Atari BASIC commands, SAVE, LOAD, LIST and ENTER must be used to store and retrieve a wedged Voice Master BASIC program.

```
********************
PART IV
Recognition
********************
```

This part of the manual deals exclusively with speech recognition and assumes that you have already familiarized yourself with the previous part on speech recording and playback. The same principles of assimilation and digitization of data during the recording process of speech reproduction are equally applicable to the recording process used in speech recognition. The main difference with speech recognition is that the recorded speech data must be saved and then later compared with new recorded data. When a word or phrase is recorded for recognition purposes, the 'characteristic' data of that particular word or phrase is saved in memory as a 'template'. Because a template requires much less memory for storage than does the actual recorded word or phrase, up to 32 templates can be stored in less than 4K of memory space. Hence, there is room enough in just a 48K Atari for utilization of both speech reproduction and recognition within the same BASIC program.

The TRAIN command

Words or phrases, even short utterances and noises, can be recorded and saved as templates for recognition. But, due to the time consuming calculations that are needed to recognize templates, it is impractical to make templates from words or phrases that are too long in duration. For this reason, the speech input buffer used for recognition has a limited data capacity of two seconds for all Atari models.

To record data for recognition, the 'TRAIN n' command is used. This command is analogous to the 'LEARN n' command and is used in the same way. The parameter 'n' however, has a range of only 0 to 31. To TRAIN a word or phrase just type 'TRAIN n', with the appropriate value for 'n' and press RETURN. When you do this, an inverse question mark, '?', will appear in the prompt window. As with the LEARN command, you may pause the record routine at this point by pressing the up arrow '^' or exit the routine by pressing BREAK (or any other key). In fact, because the same machine language recording routine is used, keyboard handling and 'time-out' operate identically for TRAIN and RECOG (described below) as they do for the LEARN command. After the TRAIN command has been issued, just speak a word or phrase into the microphone and it will be recorded and saved as a template. If the computer 'beeps' while TRAINing a word, then the TRAIN was 'bad' and the word must be re-TRAINed. If the computer 'beeps' again when the word is re-TRAINed, then the template for the word may have to be BLANKed (see below) before the TRAIN will be successful. For the word to be recognized, it should be TRAINed twice — once may not be sufficient. When a word or phrase with the same template number, 'n', is re-TRAINed, its 'characteristics' are averaged with those of the previously saved template and a 'new', modified template is saved in the 'old' one's place.

An average can be acquired for any number of repetitions of the same word or phrase which has the same template number. However, TRAINing a word or phrase more than twice will probably not improve the template and might actually degrade it.

The BLANK command

There are times when you will need to erase a template from memory. The 'BLANK n' command will erase the specific template (**0 to 31**) designated by its index number, 'n'. No other templates in memory will be erased if an index number is specified. <u>If the BLANK command is entered alone, without a template number, then all templates stored in memory will be BLANKed.</u> BLANK and CLEAR, without the index parameter, are analogous in their function.

The RECOG command

Once the desired number of templates has been made with the TRAIN command, then words or phrases can be recognized. The RECOG command is used to record a word or phrase that is to be recognized. Simply type RECOG (without any parameters) and press RETURN. Again, as with the TRAIN command, an inverse question mark will appear in the prompt window. You can also pause or exit the routine as described for the LEARN command. Now speak one of the words or phrases that was previously TRAINed. If the word or phrase is recognized, a value corresponding to the number of its template will be stored in memory LOCATION 209. Enter PRINT PEEK(209) to examine the contents of the location. If any number other than the correct template number for the word or phrase RECOGnized is found here, then the word or phrase was incorrectly recognized or the computer refused to make an identification. See APPENDIX III for a complete table of values for this location. Here is a BASIC program which exemplifies use of the TRAIN, BLANK and RECOG commands:

```
10 FOR NUMBER=1 TO 5
20 PRINT "Please say the number: ";NUMBER
30 TRAIN NUMBER
40 IF PEEK(209)=248 THEN 140 :REM Check for 'bad' train error.
50 NEXT NUMBER :REM If no error, then train next template.
60 PRINT CHR$(125) :Clear the screen.
70 PRINT "Say a number between 1 and 5." :REM Prompt to do a RECOG.
80 RECOG :RESULT=PEEK(209) :REM Do recognition and get result.
90 IF RESULT<1 OR RESULT>5 THEN 120 :REM Was number recognized?
100 PRINT "You said: ";NUMBER
110 GOTO 80 :REM Go do another recognition.
120 PRINT "Word not recognized! Please try again."
130 GOTO 80 :Go try again.
140 PRINT "BAD TRAIN!"
150 BLANK NUMBER :Erase the offending template.
160 POKE 209,128:GOTO 30 :REM Acknowledge error and re-train.
```

Template partitioning

The algorithm used to compare a template with a word or phrase to be recognized is quite complex. So a considerable amount of microprocessor cycles are required to scan all the templates and find a comparison. And though it takes less than a second to scan all 32 templates, there are programming applications where a minimum amount of scanning time is desirable. This is one reason that recognition can be done via partitioning. That is, the RECOG command can be used with a field of parameters that specifies which part, or **subgroup**, of the entire group of 32 templates, is to be scanned.

The format for the command is 'RECOG a,b,c,d', where the variables a,b,c **and d**, are independently optional and can be assigned any value from 0 to 4. The **subgroup(s)** of templates that will be scanned when a parameter is specified is as follows:

RECOG 0 Scans all 4 subgroups or 32 templates; same as RECOG
RECOG 1 Scans subgroup 1 or templates 0-7 only.
RECOG 2 Scans subgroup 2 or templates 8-15 only.
RECOG 3 Scans subgroup 3 or templates 16-23 only.
RECOG 4 Scans subgroup 4 or templates 24-31 only.

One to four subgroups of templates can be scanned and in any order desirable. Some examples are:

RECOG 3,1 ;this command will scan templates 16-23 first and then scan templates 0-7.
RECOG 1,3,2 ;this scans templates 0-7, 16-23 and 8-15 respectively.
RECOG 4,3,2,1 ;this scans all subgroups of templates in reverse order.

Aside from reducing the amount of scan time required to do a recognition, template partitioning can also be used to improve accuracy when using categorized vocabularies. For example, suppose we set up a restaurant menu with three main categories, meals, desserts and beverages. We then TRAIN the three names of the categories to the first subgroup of templates. Further, we TRAIN the second subgroup with items belonging to the category of meals, the third subgroup with desert items and the fourth with beverage items. Now we do a RECOG1 and say, "beverages". The word is recognized and the result of the recognition is used to branch the program to a RECOG4, where we can now say what particular beverage we desire. We say, "beer!". Of course, 'pull-down' type menus could have been used to display categories and items.

Recognition and the SCREEN command

In addition to using template partitioning, a SCREEN 3 command is available which can be used to reduce the scan time by an additional 30%. When SCREEN 3 has been entered, the screen will go blank only while the templates are being scanned. To undue the SCREEN 3 command, just enter SCREEN 0.

As with the LEARN command, the screen can also be turned off while recording speech with the TRAIN and RECOG commands. Enter SCREEN 1 to turn it off, and SCREEN 0 to turn it back on again. In some cases turning the screen off may help improve recognition accuracy. However, if the screen is turned off to TRAIN a template, it must also be off when the same template is RECOGed.

The UNBLANK command

The 'UNBLANK n' command, where 'n' specifies the particular template (0 to 31) to be UNBLANKed, will restore a template that has been previously 'BLANKed'. Like the BLANK command, entering UNBLANK without a parameter will UNBLANK all the templates. There are actually two uses for this command. The first, and obvious use, is to restore templates that have be accidentally erased with the BLANK command. The second, less apparent use, is to enhance recognition accuracy by allowing selective recognition of templates. This is similar in concept to

the partitioning method described above, except that, this method can be selective even within a particular subgroup of templates. For example, suppose we train eight words to the first subgroup of templates. Now suppose that we want to consider only four of the eight templates when we do the RECOGnition. We first BLANK the four templates that we don't want scanned, then we do a RECOG1. Only the templates that were not BLANK are considered as a possible candidate for a match and the UNBLANKed ones are ignored. Finally, the previously BLANKed templates can be restored with the UNBLANK command, so that all eight templates will again be available for RECOGnizing. This method effectively increases recognition accuracy by eliminating, from the selection process, those templates which might otherwise be wrongly chosen by the RECOGnition routine as the correct match.

Sensitivity thresholds

When a word or phrase is compared to a TRAINed template with the RECOG command, it is compared to all of the templates (within the selected template subgroups) and a numerical score is computed for each comparison. These scores are then used to either accept or reject the RECOGed word based on two major criteria, the 'maximum error score' and the 'minimum error score'.

The maximum error score is derived from the number of dissimilar 'characteristics' found between the word being RECOGnized and the one template which most closely matches it. The smaller the computed value for the score, then the fewer are the dissimilarities; and the closer the word or phrase matches the selected template. If the maximum score is greater than the preset maximum threshold, then the RECOGed word is rejected.

The minimum error score is derived from the number similar 'characteristics' found between the two templates which most closely match the word being RECOGnized. This is determined by computing the difference between the template which most closely matched the RECOGed word and the template which was the second closest match. If this difference is greater than the preset minimum threshold, then the RECOGed word is rejected.

Several locations have been set aside in memory for storing the results of the above computations as well as the values for the maximum and minimum threshold settings. They are:

LOCATION	FUNCTION
1648	Index number of closest match.
1649,1650	Error score for closest match (Low/High bytes).
1651	Index number of second closest match.
1652,1653	Error score for second closest match (L/H bytes).
1654,1655	Minimum threshold value (L/H bytes).
1656,1657	Maximum threshold value (L/H bytes).

The above locations can be PEEKed after a RECOG to examine the values contained in them.

The ACCEPT command

The 'ACCEPT n' command, where 'n' specifies one of five acceptance levels from 0 to 4, is used to decrease or increase the 'sensitivity' of the RECOG command. Lowering the ACCEPTance level causes the recognition routine to be less selective when looking for a match. In other words, a RECOGnition will be less sensitive to discrepancies and variations between the word being RECOGnized and the template that is selected as its match. Entering ACCEPT0 will allow RECOGnition of almost any word without returning an error. However, this also increases the likelyhood that an incorrect match will be made. Raising the ACCEPTance level has the opposite effect.

The ACCEPT command functions by altering the maximum and minimum threshold settings, in accordance with the following:

n	Min	Max
0	1	250
1	5	200
2	10	150
3	15	100
4	20	100

The default condition for the ACCEPT command is n=2 and should be adequate for most purposes, but feel free to experiment with different settings.

The TSAVE & TLOAD commands

TRAINed templates can be saved to and loaded from disk or cassette tape with the TSAVE"D:FILENAME" and TLOAD"D:FILENAME" commands. See "APPENDIX II" for a detailed explanation of the template file structure. As with speech files, there is no theoretical limit to the number of templates that can be TSAVEd for later retrieval and use by your BASIC program.

Hints and techniques

No computer has the capability to recognize words as well as a human being. In computer terminology, the human brain has several distinct advantages over the average computer such as, several hundred billion bytes of memory, parallel processing capability, and advanced Artificial Intelligence (A.I.) routines that took untold millennia to evolve. Considering this comparison, it seems truly amazing that the Voice Master performs as well as it does! So don't ask the Voice Master to recognize between words and sounds that even a human would have difficulty with. Think how often, when listening over a telephone, you must have the spelling of a name or a number repeated so that you can recognize it. Can you expect the computer to do better?

When working with recognition it is very important to practice good microphone technique (as was described in PART I). The Voice Master contains an automatic volume control mechanism, but it is not as versatile as the equivalent in the human ear. So, attempt to say your words at an adequate volume level to get well above any background noise. In addition, try to maintain consistency while speaking, so that a RECOGed word sounds the same to the Voice Master as

when the word was TRAINed. But, try not to 'go monotone'. A natural, unstrained voice works best. Be aware that your voice probably varies somewhat in accordance with mood, environment and other factors. So, if you are attempting to do recognitions from templates that were previously made and TSAVEd to disk when you had a cold, then you will likely experience some disappointment.

Multi-syllable words and words which do not sound similar to one another are most easily recognized. Letters like 'B' and 'D' are very difficult for the Voice Master to distinguish between. However, words 'Bravo' and 'Delta' are easily recognized. Keep this in mind when you design a recognition vocabulary and try to choose words that will be easily distinguishable from one another. Because recognizing letters and numbers can be difficult, an 'International Phonetic Alphabet' is provided in APPENDIX VI. We suggest that you use the phonetic alphabet for recognizing letters with the Voice Master.

Although recognition templates are generally restricted to use by the particular person who TRAINed them, some word sounds are more adaptable between different speakers than are others, and special 'multi-speaker' vocabularies can be constructed. However, the technology for multi-speaker speech recognition is still in the infancy stage, so the success rate in this area will be minimal. However, we do encourage experimentation.

The ACCEPT command is the easiest way to alter the error thresholds and is adequate for almost all applications. However, if you wish to alter the minimum and maximum error thresholds directly, then simply POKE the appropriate values into their respective memory locations and they will be used for all succeeding RECOGnitions. If the templates are later TSAVEd, the values last POKEd will be saved also.

Because a subgroup of templates consists of a minimum of 8 templates, all un-TRAINed templates in the subgroup should be BLANK to prevent accidental false recognition. For example, if there are, say, 6 words which have TRAINed templates in memory, then the scan will cover two additional templates. These two templates should be BLANKed if they were previously used (TRAINed).

```
**************************************************
                    PART V
         Loading And Using The Voice Master Programs
**************************************************
```

In this part of the manual, each individual program on the Voice Master disk will be described in detail. Additionally, instructions will be given on how to make a customized disk containing only those programs that you need for your particular application.

The most important program

The main Voice Master program, hereinafter referred to as the 'MASTER' program, is the one that 'drives' the hardware, analyzes and stores in memory speech and recognition data, and performs all the supplementary data management chores. It even manages the wedge which gives you easy access Voice Master functions via added BASIC commands. If this program is not loaded into memory and running, then you will not be able to record or recognize words and phrases or use the wedged commands. There are three separate versions of the MASTER program. These are:

> **"VM.800"** for the Atari 800 or 1200XL
> **"VM.XL"** for the Atari 800XL or 65XE
> **"VM.XE"** for the Atari 130XE

Each program performs all the same general functions as the others, but differs from them in what memory locations it uses to input and store speech and recognition data. For example, the VM.XE version stores speech data in the extra 64K bank of memory, the other two versions do not (see APPENDIX I for more information on memory usage). This is because the VM.XE version was designed to take advantage of the extra memory that is available on the Atari 130XE and will not run on the Atari 800 or 800XL. Similarly, the VM.XL version will not run on the Atari 800.

However, the MASTER programs are upward compatible and the VM.800 version of the program will run on the Atari 800XL and Atari 130XE, as well as the Atari 800. Likewise, the VM.XL version will run on the Atari 130XE as well as the Atari 800XL model.

Also, speech files created with the VM.XE version and SSAVEd cannot be SLOADed and played back with one of the other two versions. But, like the MASTER programs themselves, speech files are also upward compatible. So, a speech file made with the VM.800 version, for example, can be SLOADed and played back with the VM.XL or VM.XE versions.

On the other hand, recognition templates are NOT upward compatible and you cannot TLOAD and RECOG templates with the VM.XE or VM.XL versions that were TRAINed and TSAVEd with the VM.800 version. Or vice versa.

All three MASTER program versions are binary files and require an associated 'loader' program to place them in memory. The loader programs are:

```
"VM800"  for the VM.800 version
"VMXL"   for the VM.XL version
"VMXE"   for the VM.XE version
```

These programs are written in BASIC and can be RUN by entering at the READY prompt, one of the following appropriate commands:

```
RUN"D:VM800"
RUN"D:VMXL"
RUN"D:VMXE"
```

When the loader program has finished doing its job, the corresponding MASTER program will be resident in memory, initialized, protected from system RESET, and ready for use. Since the program is 'reset proof', neither it or any other data, including a BASIC program, will be lost if you need to press RESET for some reason.

NOTE: Although pressing RESET does not cause data to be lost, all of the MASTER program pointers, as well as BASIC's pointers (except RAMTOP), are restored to their default values.

Each version of the MASTER program reserves a specified amount of computer memory for BASIC programs. Enter FRE(0) to see how much memory is available for this purpose. If you need more memory, then use the CLEAR command as described in PART II of the manual. If an attempt is made to load a BASIC program that is too large to fit in the reserved memory, then an ERROR-19 will be returned. If this happens, it means that you must CLEAR more memory before the BASIC program will load.

IMPORTANT NOTE: If you are using the VM.800 version, at least 64 pages must be CLEARed before accessing any DOS function or some BASIC programs on the Voice Master disk. Just enter CLEAR 64. (Note: the "MENU" program automatically does this for you when the 'EXIT TO BASIC' option is selected.)

The "MENU" program

This program is responsible for loading, running and allowing easy access to the various programs that are on the Voice Master disk. The "MENU" program automatically loads and runs (auto-boots) via an "AUTORUN.SYS" file. In turn, the "MENU" program loads in the MASTER program applicable to your model of Atari (800, XL or XE) and initializes it. Although convenient, use of the "MENU" program is not mandatory since all the programs, including "MENU" itself, can be loaded and 'RUN' independently. If you do not want the "MENU" program to auto-boot, then simply change the name of the "AUTORUN.SYS" file to "AUTORUN" according to the directions below:

1. Boot-up the copy you made of the Voice Master disk according to the directions given in PART I of the manual.

2. From the Voice Master MENU, select the 'EXIT TO BASIC' option and type DOS at the READY prompt, then press RETURN

3. When the DOS menu appears on screen, select item 'E. RENAME FILE'.

When you see the prompt: 'RENAME – GIVE OLD NAME, NEW'.
Answer by typing: 'AUTORUN.SYS,AUTORUN' and then press RETURN.

This will change the name of the file to "AUTORUN", which becomes inactive without the '.SYS' extender. When the computer boots-up into BASIC, control will be given over to you instead of the "MENU" program. The "MENU" program can still be used, at any time, by entering RUN"D:MENU". To restore the auto-boot function, just rename the file back to "AUTORUN.SYS".

The "BAR.COM" program

This is a binary file that can be accessed from menu or loaded directly from DOS. The program is designed to work either independently from, or in conjunction with, the MASTER program. Follow the steps below to load the "BAR.COM" program:

1. Boot-up the copy you made of the Voice Master disk according to the directions given in PART I and select the 'EXIT TO BASIC' option from the menu.

2. Type 'POKE 106,144:GRAPHICS 0' and press RETURN. NOTE: If the VM.800 or VM.XL MASTER program is resident, skip this step.

3. Type 'DOS' at the READY prompt, then press RETURN

4. When the DOS menu appears on screen, select item 'L. BINARY LOAD'.
When you see the prompt: 'LOAD FROM WHAT FILE?'.
Answer by typing: 'BAR.COM' and then press RETURN.

When loaded, the "BAR.COM" program occupies the area of memory from decimal address 39936 to address 40959, which happens to be the same area of memory occupied by the Atari's display list and screen display data. Step #2 above, lowers the value of RAMTOP (location #106) to 144 and moves the display list and screen data below the new value so that the area above 39936 will be clear for loading in the "BAR.COM" program.

A complete description of the bar-display was given in PART I of the manual, please refer to it if necessary.

To exit the "BAR.COM" program and return to DOS, just press the SPACEBAR or any other key. If you want to return to BASIC from DOS, then select item: 'B. RUN CARTRIDGE'. From BASIC you can restart the "BAR.COM" program again by entering BAR=USR(39936).

NOTE 1: Pressing RESET will cause the "BAR.COM" program to be 'wiped out'. If this happens, you will need to re-load it.

NOTE 2: Unlike the MASTER program which can be configured to use either joystick port #1 or port #2, the "BAR.COM" program only works with the Voice Master unit plugged into PORT 2.

The "PLAY" program

One of the many good reasons for owning a Voice Master is that you now have the capability to add speech to any of your BASIC programs. And best of all, you

do not need to have the Voice Master hardware 'hooked-up' to play back the speech. All you need is this little (about 500 bytes) "PLAY" program and the POKEY sound chip that is already built into your Atari. The "PLAY" program contains data that was acquired from a 'relocatable', binary file that could load and play back speech files created with the MASTER program. When the program is initialized, the binary data is assigned to a string variable called 'ML$', which is stored in BASIC program memory. The first memory location of 'ML$' is the beginning of the binary data for the Machine Language routine. Because "PLAY" is written in BASIC, it can easily be merged with your BASIC program as follows:

1. Before merging the two programs, be sure that your BASIC program does not use line numbers above 31999.

2. Boot-up the disk containing the BASIC program that you want to add speech to. Then LOAD (or ENTER) your BASIC program and remove the disk.

3. Insert the disk with the "PLAY" program on it.

4. Type: ENTER"D:PLAY" and press RETURN. Then remove the disk.

5. Insert the disk on which you want to save the merged program and SAVE it with a filename of your choice.

To use the "PLAY" program, you must first create a speech file with the MASTER program and SSAVE it to disk with the filename, "D:SPEECH". Of course, other filenames can be used and you may access more than one speech file with the program. Just LIST the "PLAY" program to the screen and make the appropriate changes. When making the speech file, do NOT use the VM.XE MASTER program or "PLAY" will not be able to SLOAD or playback the speech file. This is because "PLAY" cannot access the extra 64K bank on an Atari 130XE. You must use either the VM.800 or the VM.XL version of the MASTER program to make the speech file for "PLAY". Therefore, if you are adding speech to a BASIC program that you want to work with any model of Atari (a commercial application perhaps), then you should create the speech file with the VM.800 version. For further instructions on using the "PLAY" program, ENTER and LIST the program to the screen and read the REM statements.

The "EDITOR" program

The speech "EDITOR" program can be used to edit the amplitude data of a speech file. Amplitude editing can significantly improve the sound quality of speech during playback. The speech editor is not difficult to use, but does require some preliminary familiarity with the Voice Master. We suggest that you read the manual and practice with the Voice Master before using the speech editor.

The "EDITOR" program is written in BASIC and can be activated by selecting it from the main menu or from BASIC by typing RUN"D:EDITOR". When the title screen appears, press any key and an editing window will be displayed. Notice that the editing window is 'empty' except for the graduated red cursor at the left side.

There are two ways to get speech data into memory and displayed in the editing window. One way is to select the 'LEARN WORD' option from the editor menu, answer the prompts and press RETURN after each answer then directly enter the speech data via the microphone. After the word has been recorded (LEARNed), the speech amplitude data will be plotted in the editing window as series of light-blue dots, where each dot represents a single byte of amplitude note. Because the Voice Master Jr. has no amplitude input circuitry, all the dots will initially be set to a value of 15 after recording or loading speech. This is normal.

The other way to get speech data in memory is to select the 'GET SPEECH FILE' option. The program will then prompt you to enter the name of a speech data file. Just type the filename (e.g 'DEALER.SPE') then press RETURN and the speech file will be loaded from disk into memory. Select the 'EDIT NEW WORD' option to let the program know which word of the speech file you want to edit and the speech amplitude data will then appear in the editing window.

Editing the speech data is easily done using three pairs of keys: The first pair of keys, less than,'<' and greater than, '>', will move the editing window, left or right respectively, over the entire length of the speech data. Thus, any portion of a word can be centered within the window for editing. The four cursor control keys (arrows) which are normally used for screen editing are also used for speech editing, except that, it is not necessary to hold down the CONTROL key. Just press the arrow keys. The pair of left and right arrow keys will move the editing cursor across the speech data so that any individual amplitude byte (blue dot) can be selected for editing. The pair of up and down arrow keys are used to raise or lower the amplitude byte as desired.
Move the amplitude byte up to give it more volume and down for less. Lowering the amplitude byte down to zero, or the bottom of the cursor, will turn the amplitude completely off for that portion of the speech.

To hear what the edited speech sounds like, select the SPEAK WORD option and the entire word will playback. To play back just a small portion of the word, use the '<' and '>' keys to position the left frame of the editing window just in front of the first amplitude byte where you want speech playback to start from. Now use the left or right arrow key to position the editing cursor on the amplitude byte where you want the playback to stop and then select the CURSOR PLAYBACK option. This option can be used to playback any portion of a word which will fit within the editing window.

Sometimes a word can be made to sound worse by over-editing or editing the wrong amplitude bytes. If this happens, the 'RESTORE WORD' option will restore all amplitude bytes to their original values.

When you have finished editing a word in a speech file, another word in the same file can be edited by selecting the 'EDIT NEW WORD' option and answering the prompt. The speech data for the next word will appear in the editing window and the changes made to the previous word will be retained. This can be done for every word in the speech file if desired. When you have finished editing, use the 'PUT SPEECH FILE' option to save the edited vocabulary back to disk.

Finally, when you have finished with the editing program, use the 'TO VM MAIN MENU' option to exit the program and return to the main menu or press RESET to return to BASIC.

Unfortunately, there are no set guidelines for speech editing and you will just have to experiment with various sounds until you acquire a 'feel' for it.

The "DEMO" program

The purpose of this program is to introduce a newcomer to the basic features of the Voice Master. It includes a speech recording and playback demonstration with options to vary the speech playback rate and speech volume (echo effect). A simple speech recognition demonstration and a speech bar-graph ("BAR.COM") program are also included. The "DEMO" program can be RUN by entering, RUN"D:DEMO", or by selecting it from the main menu.

The "CLOCK" program

With this program, you get a clock that actually speaks the time and there is an alarm mode so your Atari can wake you up in the morning. Run the program by entering, RUN"D:CLOCK" and when the introduction screen appears, just press any key to get to the CLOCK MENU. Select the 'VOICE CREATOR' option from the menu and the program will prompt you to enter a speech vocabulary. After you have done this, the program will enter the EDIT MODE. From this mode, you can SPEAK back and/or re-LEARN any individual word that was previously entered. When you have finished editing, you can save the vocabulary with the 'SAVE CLOCK VOICE' option. If you decide to save the vocabulary, then you will not have to create another clock voice the next time you run the program. When the program finally returns to the CLOCK MENU, just select the 'SET CLOCK TIME' option, follow the prompts, and the clock will start ticking.

The "BLACKJACK" program

"Welcome folks, to the Voice Master talking Blackjack game! Just step right up and place your bets please...here's your opportunity to 'hit the big one' and the house will give you twenty bucks to get started with!" Want to play? The rules are simple:

1. With the MASTER program resident, enter: RUN"D:BLACKJACK" (See the note below). Or, select the program from the main Voice Master MENU.

2. Follow the prompts and speak, very clearly and distinctly, the words that the program requests you to say. When all the templates have been made, you can start playing the game.

3. The minimum bet allowed is two dollars. When asked to place your bet, enter it one number at a time. If you make a mistake, then say: 'ERASE' and enter the numbers again. When you have finished entering your bet, say: 'BET'. The cards will then be dealt out. The object of the game is to get a higher value hand then the dealer's without exceeding ('busting') a total value of 21. Face cards count 10 and Aces count 1 or 11, as you please. Other cards count their face value. A Blackjack is any face card plus an ace.

5. When both hands have been dealt you will be prompted to make a choice. To get another card, say: 'HIT ME'. Keep taking 'hits' until you think that your hand is higher then the dealer's hand and then say: 'STAND'. Sometimes you will not need to take any hits at all, like when you get dealt a Blackjack. When this happens then just 'STAND' on it.

6. Winning hands pay 2 to 1 and Blackjack pays 3 to 1. A 'push' (both hands equal) pays 1 to 1, which means that you don't lose your bet. Also, if you get more than five cards in your hand without 'going bust', a five-card Charlie', then you win.

7. The dealer must play house rules. That is, he (your Atari) must hit a

hand which equals 16 or less. He must stand on a hand which is 17 or more, unless it's a 'soft-seventeen' (a hand with an ace that counts 11).

8. Splits, double downs and insurance bets are not allowed.

If at any time during the game you want to see exactly which cards have been played, then say: 'CARDS' and you will get a display of the played cards. Say, 'CARDS', again to continue with the game. When you eventually run out of money (the 'house' always wins), then you can answer the 'Play Again?' prompt with a 'YES' to play again, or a 'NO' to return to the main VOICEMASTER MENU.

NOTE: To RUN the "BLACKJACK" program from BASIC with the VM.800 or VM.XL MASTER programs, a CLEAR 70 must be entered or else an ERROR-19 will occur.

The "BJVOICE" program

The dealer's voice supplied with the Voice Master has been highly edited using the "Speech Construction Set" and the sound quality of this speech cannot be duplicated by using the Voice Master alone. However, if you are tired of listening to the same dealer's voice in the "BLACKJACK" game, then you can easily make another with this short BASIC program. Enter: RUN"D:BJVOICE" and answer all the prompts. When you have finished entering the vocabulary, the program goes to the EDIT MODE where you can 'clean-up' any mis-LEARNed words. This edit mode is similar to the one used in the talking "CLOCK" program.

When you have finished editing the vocabulary, the program will SSAVE the file to disk with the filename: "DEALER.SPE". Be sure that the file is SSAVEd to a disk which has a copy of the "BLACKJACK" program on it and also enough extra room for the speech file itself.

NOTE: If using the VM.800 MASTER program while making a new dealer's voice, you must make your words very short or the finished vocabulary will not fit in memory.

The "COMPOSER" programs

There are two programs on the disk which will not be described here. They are: "COMPOSER" and "COMP.COM". These files constitute the famous Voice Master "Music Composer" programs and are thoroughly explained in another manual dedicated only to them.

The BASIC programs in general

All of the BASIC programs described here can be LISTed to the screen and studied, so that you can 'learn by example' how to incorporate the Voice Master commands in your own BASIC programs. You may even want to adopt some of the more useful subroutines like the vocabulary editor in "BJVOICE" or the "BAR.COM" loader found in the "DEMO" or "MENU" programs. Also, some of these programs could be edited and improved. So, feel free to edit, modify and customize them to suit your fancy.

Atari DOS 2.5 files

There are three files (programs) which are part of "ATARI DOS 2.5", these are:

```
"DOS.SYS"   the Disk Operating System
"DUP.SYS"   the Disk Utility Package
"RAMDISK"   a program for the ATARI 130XE
```

There is a manual available that completely explains the use of these, and other, DOS 2.5 files. The "ATARI DOS 2.5 OWNERS MANUAL" is available from:

```
ATARI Customer Relations
P.O. Box 61657
Sunnyvale, CA 94008
```

The price is $10.00 plus $2.50 postage.

How to make a customized Voice Master disk

As shipped, the Voice Master disk is 'stuffed' full, with hardly any room left for saving your BASIC programs and speech files. Of course, you could SAVE or SSAVE everything to a separate disk, but this tends to be inconvenient. Especially if you only have one disk drive. The solution is to make a customized version of the Voice Master disk, which has on it those programs that you need most.

If you want to make a customized disk for developing BASIC programs which include speech recording, then you must make a copy of one of the versions of the MASTER program along with its affiliated loader program. For example, if you have an ATARI 800, you will want the following programs on your customized programming disk:

```
"DOS.SYS" the Disk Operating System
"DUP.SYS" the Disk Utility Package
"VM.800"  the MASTER program
"VM800"   the affiliated loader program.
```

To make this customized disk, follow these instructions:

1. Boot-up the copy you made of the Voice Master disk according to the directions given in PART I of the manual.

2. From the 'VOICEMASTER MENU', select the 'EXIT TO BASIC' option and type DOS at the READY prompt, then press RETURN

3. Remove the Voice Master disk and insert a blank disk.

4. From the DOS menu, select item: 'P. FORMAT SINGLE'.
When you see the prompt: 'WHICH DRIVE TO FORMAT?'
Answer by typing: '1' and then press RETURN.
When you see the prompt: 'TYPE "Y" TO FORMAT DISK 1'.
Answer by typing: 'Y' and then press RETURN.

5. From the DOS menu, select item: 'H. WRITE DOS FILES'
When you see the prompt: 'DRIVE TO WRITE DOS FILES TO?'
Answer by typing: '1' and then press RETURN

When you see the prompt: 'TYPE "Y" TO WRITE DOS TO DRIVE 1'.
Answer by typing: 'Y' and then press RETURN. This will write the DOS.SYS and
DUP.SYS files to the disk.

     6. Next, select from the DOS menu:
item 'O. DUPLICATE FILE'.
When you see the prompt:'NAME OF FILE TO MOVE?'
Answer by typing: the filename of a Voice Master file that you want copied on to
the new disk.
When you see the prompt: 'INSERT SOURCE DISK,TYPE RETURN'.
Answer by inserting the Voice Master disk, then pressing RETURN.
When you see the prompt:
'INSERT DESTINATION DISK,TYPE RETURN'.
Answer by removing the Voice Master disk and inserting the blank, formatted
disk, then pressing RETURN. Repeat this step for each file that you want to put
on your customized disk.

     If you have an ATARI 130XE, then you have the option of using the extra 64K
bank of memory for either speech storage or as a 'ramdisk'. To use the ramdisk,
make a copy of the VM.XL version of the MASTER program instead of the VM.XE
version and include a copy of the "RAMDISK" file. Rename the "RAMDISK" file to,
"RAMDISK.COM" in order to activate it. To rename the file, just follow the same
instructions given for renaming the "AUTORUN.SYS" file.

Speech data alone would be useless without an orderly method of referencing it. So, the MASTER program reserves the first 330 bytes of the speech storage area for storing reference 'pointers' and the remaining area is for the speech data itself. Both the pointers and speech data together constitute a 'speech file'. The very first location of the speech file is known as the 'BASE' address. The BASE is set to decimal address 16384 ($4000) by default, but can be reset to another memory location by use of the CLEAR command. The method for determining the BASE address depends on which version of the MASTER program you are using.

If you are using the VM.800 version, the BASE address is: BASE=(PEEK(106)+16)*256. In order to accommodate the 4K speech input buffer, which is below the speech storage area, 16 pages were added to the value in location 106 (RAMTOP).

If you are using the VM.XL version, the BASE address is: BASE=PEEK(106)*256. In this case, the speech storage area begins at the BASE address.

If you are using the VM.XE version, the BASE address is: BASE=64*256. This is always the case because the speech file is stored in the extra 64K bank and must be accessed through the 'window' located from 16384 ($4000) to 32767 ($7FFF). See pp. 121-122 of your "Atari 130XE Owners Manual."

Memory locations from BASE t0 BASE+255

These first 256 bytes of memory define the starting and ending address of where a particular word or phrase is stored in memory. Two bytes are required to define the starting address of a word or phrase and two more for the ending address. So, the first two locations, BASE+0 and BASE+1, correspond to the 'low' and 'high' order bytes, respectively, for the starting address of the first word or phrase (word index #0); and the next two consecutive memory locations, BASE+2 and BASE+3, define the ending address. Therefore, for each recorded word or phrase, a total of four memory locations is needed to delimit the area of memory that is occupied by the actual speech data. To find the location of the address pointers for a word with index number 'n': **Multiply** 'n' **times 4 and add the product to the BASE address.** Then use PEEK(POINTER) to determine the location in memory of the actual speech data. The following BASIC program exemplifies this procedure:

```
10 PRINT"ENTER WORD (INDEX) NUMBER";
20 INPUT WORD
30 BASE=16384 :REM The default value is used here
40 POINTER=BASE+(WORD*4)
50 POKE 54017,225 :REM Switches 'on' extra 64K
60 START=PEEK(POINTER)+256*PEEK(POINTER+1)
70 END=PEEK(POINTER+2)+256*(POINTER+3)
80 POKE 54017,253 :REM Switches 'off' extra 64K
90 PRINT "START=";START,"END=";END
```

NOTE: This program, and others in this appendix, work the same for any model Atari. If you are not using a 130XE, then line numbers 50 and 80 (above) will have no effect. Also, it is assumed that the default BASE address has not been changed with the CLEAR command. If the BASE has been changed, then adjust line No. 30 appropriately.

Memory locations BASE+256 & BASE+257

These two memory locations define the address of the end (or top), of the entire speech file. The first location is the low order byte and the second location is the high order byte of the address.

To find out how much total memory the speech file occupies, use the following program:

```
10 POKE 54017,225
20 BASE=16384
30 TOPMEM=BASE+256
40 PRINT (PEEK(TOPMEM)+256*PEEK(TOPMEM+1)-BASE)
50 POKE 54017,253
```

NOTE: When a speech file is SSAVEd, all speech data, from BASE to the address specified by TOPMEM, is saved to disk or tape.

Memory location BASE+259

This location contains the count of the total number of words or phrases that have been LEARNed (recorded).

Memory location BASE+264

This is the 'flag' location for the playback screen. A value of 34 indicates the normal 'screen on' condition. A zero indicates the screen will be off during playback.

Memory location BASE+265

This is the 'flag' location for the recording rate. Three recording rates are available and the numerical value found in this location corresponds to the rates as follows:

VALUE	RATE
1	15,600 Khz
2	7,800 Khz
4	3,900 Khz

The values given for the recording rates can be used to calculate the amount of memory consumed by speech data if the time factor is known. The time, in seconds, can be measured with a stop-watch; using 'SPEAK n' to playback the speech. The equation is:

$$BYTES=Khz*SECONDS/8 \text{ BITS per second}$$

For example, a word recorded at RATE2 that is measured to be 10 seconds in duration, requires: 7,800*10/8 = 9750 BYTES.

Conversely, the amount of playback time available for a given amount of memory can be calculated as follows:

$$TIME=BYTES*8/Khz$$

For example, the amount of time available for storing speech data in the extra 64K bank of the Atari 130XE is calculated by first subtracting the bytes used by the speech file pointers from the total amount of available bytes: 65,535-331 = 65,205 BYTES. Then:65,205*8/7,800 = 66 SECONDS or 1 minute and 6 seconds.

Memory location BASE+266

This is the 'flag' location for the playback rate. Five playback rates are available and the values found in this location correspond to the rates as follows:

VALUE	RATE
0	30,000 Khz (Approx.)
1	15,600 Khz
2	7,800 Khz
3	5,200 Khz
4	3,900 Khz

Memory locations BASE+267 to BASE+330

These 64 bytes are an index of the order which the words or phrases have been recorded in. For example, if you LEARNed phrases 3,8,2,0,40 and 23, in that order, then the first six memory locations, beginning at BASE+267, will contain 3,8,2,0,40 and 23, respectively. BASE+267 will be 3, BASE+268 will be 8 and so on. The following BASIC program checks these pointers to see if a particular word has been recorded:

```
10 POKE 54017,225
20 BASE=16384
30 PRINT"ENTER WORD INDEX NUMBER FOR SEARCH";
40 INPUT WORDNUM
50 WORDCNT=PEEK(BASE+259)
60 FOR X=0 TO WORDCNT-1
70 ORDERNUM=PEEK(BASE+267+X)
80 IF ORDERNUM=WORDNUM THEN PRINT"FOUND!":GOTO 110
90 NEXT X
100 PRINT"NOT FOUND."
110 END
```

Memory locations BASE+331 to end of speech file

Here is where the actual speech data bits are stored. The amount of memory available for storage of speech data depends on the version of the MASTER program in use.

If using the VM.800 version, a maximum length speech file will occupy memory locations BASE+331 to 30207 ($75FF). If the FREE command was entered, then the end of the speech file is extended to 34303 ($85FF).

If using the VM.XL version, a maximum length speech file will occupy memory locations BASE+331 to 39935 ($9BFF).

With the VM.XE version, the speech file is actually stored in the extra 64K bank which has an address range of 0 ($0000) to 65,535 ($FFFF). However, the speech file can only be examined through the 16K 'window' within the address range 16384 ($4000) to 32767 ($7FFF). To examine the speech file pointers and the beginning of the speech data, first POKE location 54017 with 225. This will expose the first 16K block of the 64K extra bank. To expose the second, third and fourth 16K blocks, POKE 54017 with 229, 233 and 237 respectively. In this manner the entire speech file can be examined. Remember to POKE 54017 with 253 when you have finished examining the file. This will restore the 'window' to its default value.

Speech data bytes for any particular word are stored linearly in memory. The very first byte of the word data is the value for the amplitude 'step counter'. Since it is not necessary to sample amplitude as often as frequency, a step counter determines the sampling ratio of amplitude to frequency; or how many frequency samples are taken before sampling amplitude. The step counter varies in value in accordance with the recording rate as follows:RATE1 = 8, RATE2 = 16, RATE3 = 32. This ensures that an amplitude byte is sampled every 2.0 milliseconds, no matter which recording RATE is used. The playback routine must 'know' the value of the step counter, for any particular word, so that it can SPEAK the word back at the correct SPEED. Here is an illustration of how the speech data for a word is arranged in memory:

step counter	Amp. Byte	8,16 or 32 Freq. bytes	Amp. Byte	8,16 or 32 Freq. bytes	....

Note: The Voice Master Jr. does not sample amplitude, but sets each amplitude byte to a maximum value of 15 instead. Of course, you can modify amplitude values for playback using the amplitude editor.

```
**************************
         APPENDIX II
      Recognition Files
**************************
```

A recognition data file consists of 32 word templates plus 4 bytes for saving the minimum and maximum threshold settings. Since each template is 100 bytes long, all 32 templates require a total of 3,200 bytes of memory for storage. However, when the file is TSAVEd to disk, it is saved as a 4K block. All templates are saved, regardless of whether some of the templates were TRAINed or not.

If using the VM.800 version of the MASTER program, the recognition file will occupy the area of memory from 30208 ($7600) to 34303 ($85FF).

If using the VM.XL or VM.XE version of the MASTER program, the recognition templates are stored behind the 'Operating System' ROM and cannot be examined from BASIC. They occupy the area of memory from 49152 ($C000) to 53247 ($CFFF). Templates made from these two versions of the MASTER program are identical and can be used interchangeably with one another, but not with the VM.800 version.

Template structure

Each template of 100 bytes can be sub-divided into a 96 byte (8 by 12) grid plus a 4 byte prefix. When a word is recorded for a TRAIN, the recognition routine must either 'shrink' or 'expand' the length of the word so that it will fit into the grid of the template. So, short words like "at" are 'stretched out' and longer phrases like, "Voice Master", are 'shrunk down'. This process is known as 'time normalization' and allows for words or phrases of various lengths to be standardized into a set pattern of twelve time slots, each eight bytes wide. This makes it much simpler to compare words of various duration.

The 4 byte prefix for each template is used to store various 'flags', though only the first and second byte are currently used. The very first byte is used to indicate the BLANKed (0) or UNBLANKed (1) state of the template. The second byte is for internal use by the TRAIN routine. If you want to use BASIC to examine the templates yourself, then feel free to do so. Just remember to use the VM.800 version of the MASTER programs.

```
******************************
         APPENDIX III
       Values Associated With
      Memory Location #209
******************************
```

VALUE   EXPLANATION

0-31   The recognition template numbers for recognized words.

128   The default value when no error or recognition has occurred.
           POKE this number into 209 to acknowledge an error.

248   The TRAIN was 'bad' and must be repeated.

249   The 'SPEAK n' command was given for an unLEARNed word.

250   A 'time-out' has occurred.

251   The record or playback routine was exited prematurely by pressing a
           key during a LEARN, TRAIN, RECOG or SPEAK command.

252   The speech storage area is full. No more new words can be LEARNed.

253   The speech input buffer was exceeded during the previous LEARN,
           TRAIN, or RECOG command.

254   The word was not RECOGnized due to minimum error rejection; two or
           more words were too similar.

255   The word was not RECOGnized due to maximum error rejection; no word
           was close enough to qualify.

```
*************************************
              APPENDIX IV
      Voice Master Memory Usage Maps
*************************************
```

Map for VM.800 MASTER program
```
                                          _65535 ($FFFF)
  _____
 | 24K    OPERATING SYSTEM            |
 |          AND BASIC ROM             |
 | THIS AREA IS NOT USED BY VM.800    |
 |                                    |_40960 ($A000)
 |_____|
 | 1K      FREE RAM SPACE             |
 |    ('BARGRAPH' loads here.)        |
 |                                    |_39936 ($9C00)
 |_____|
 | 5K+       MASTER PROGRAM           |
 |             VM.800                 |
 |                                    |_34304 ($8600)
 |_____|
 | 4K    RECOGNITION DATA FILE        |
 |                                    |_30208 ($7600)
 |_____|
 | 13K+   SPEECH DATA FILE            |
 |   (size varies with CLEAR cmd.)    |
 | - - - - - - - - - - - - - - -      |-16714 ($414B)
 |    POINTERS FOR SPEECH FILE        |
 |                                    |_16384 ($4000)
 |_____|
 | 4K     SPEECH INPUT BUFFER         |
 |                                    |_12288 ($3000)
 |_____|
 |   DISPLAY LIST AND SCREEN DATA     |
 |           960 BYTES                |
 |                                    |_11264 ($2C00)
 |_____|
 | 3.5K+   FREE RAM SPACE             |
 |   FOR BASIC AND BASIC PROGRAMS     |
 |                                    |_7420 ($1CFC)
 |_____|
 | 5.5K+   DOS IF PRESENT             |
 |                                    |_1792 ($700)
 |_____|
 |     PROGRAM JUMP TABLE             |
 |     AND SPECIAL STORAGE            |
 |                                    |_1648 ($670)
 |_____|
 |       FREE RAM SPACE               |
 |                                    |_1536 ($600)
 |_____|
 |   STACK USE AT LOCATION 256        |
 |                                    |_256 ($100)
 |_____|
 |  PROGRAM ZP FROM 207 TO 255        |
 | - - - - - - - - - - - - - - -      |-207 ($CF)
 |  FREE ZP FROM 203 THROUGH 206      |
 | - - - - - - - - - - - - - - -      |-203 ($CB)
 |         OS ZERO PAGE               |
 |                                    |_0
 |_____|
```

Map for VM.XL MASTER program

```
                                              65535 ($FFFF)
|---------------------------------------|
| 7K    SPEECH INPUT BUFFER             |
|                                       |
|                                       | 58368 ($E400)
|---------------------------------------|
| 1K     CHARACTER SET                  |
|       (COPIED FROM ROM)               |
|                                       | 57344 ($E000)
|---------------------------------------|
| 2K      FREE RAM SPACE                |
|                                       | 55296 ($D800)
|---------------------------------------|
| 2K    HARDWARE REGISTERS              |
|                                       | 53248 ($D000)
|---------------------------------------|
| 4K    RECOGNITION DATA FILE           |
|                                       | 49152 ($C000)
|---------------------------------------|
| 8K     RAM BEHIND BASIC ROM           |
|      VM.XL MASTER PROGRAM (7K)        |
|          RESIDES HERE                 |
|                                       |
|                                       | 40960 ($A000)
|---------------------------------------|
| 1K     FREE RAM SPACE                 |
|      ('BARGRAPH' loads here.)         |
|                                       | 39936 ($9C00)
|---------------------------------------|
| 23K+   SPEECH DATA FILE               |
| (size varies with CLEAR cmd.)         |
| - - - - - - - - - - - - - - - -       |-16714 ($414B)
|     POINTERS FOR SPEECH FILE          |
|                                       | 16384 ($4000)
|---------------------------------------|
|   DISPLAY LIST AND SCREEN DATA        |
|           960 BYTES                   |
|                                       | 11264 ($3C00
|---------------------------------------|
| 7.5K+  FREE RAM SPACE                 |
| FOR BASIC AND BASIC PROGRAMS          |
|                                       | 7420 ($1CFC)
|---------------------------------------|
| 5.5K+   DOS IF PRESENT                |
|                                       | 1792 ($700)
|---------------------------------------|
|     PROGRAM JUMP TABLE                |
|    AND ROM SWITCH ROUTINE             |
|                                       | 1536 ($600)
|---------------------------------------|
|   STACK USE AT LOCATION 256           |
|                                       | 256 ($100)
|---------------------------------------|
|   PROGRAM ZP FROM 207 TO 255          |
| - - - - - - -  - - - - - - -          |-207 ($CF)
| FREE ZP FROM 203 THROUGH 206          |
| - - - - - - - - - - - - - - -         |-203 ($CB)
|         OS ZERO PAGE                  |
|                                       | 0
|---------------------------------------|
```

Map for VM.XE MASTER program

```
                                              65535 ($FFFF)
|‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾|
| 16K SEE THE  VM.XL MAP FOR     |
|      USAGE OF THIS AREA         |
|                                | 49152 ($C000)
|_____|
| 8K    RAM BEHIND BASIC ROM     |
|     VM.XE MASTER PROGRAM (7K)   |
|          RESIDES HERE           |
|                                | 40960 ($A000)
|_____|
|  DISPLAY LIST AND SCREEN DATA   |
|          960 BYTES              |
|   ('BARGRAPH' loads here)       |
|                                | 39936 ($9C00)
|_____|
| 32K+   FREE RAM SPACE           |
|  FOR BASIC AND BASIC PROGRAMS   |
|                                | 7420 ($1CFC)
|_____|
| 5.5K+  DOS IF PRESENT           |
|                                | 1792 ($700)
|_____|
|    PROGRAM JUMP TABLE           |
|   AND ROM SWITCH ROUTINE        |
|                                | 1536 ($600)
|_____|
|  STACK USE AT LOCATION 256      |
|                                | 256 ($100)
|_____|
|  PROGRAM ZP FROM 207 TO 255     |
|- - - - - - - - - - - - - - -   |-207 ($CF)
|  FREE ZP FROM 203 THROUGH 206   |
|- - - - - - - - - - - - - - -   |-203 ($CB)
|        OS ZERO PAGE             |
|_____| 0
```

64K speech data file in extra bank

```
                                              65535 ($FFFF)
|‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾|
| BANK4 16K SPEECH DATA    (237) |
|                                | 49152 ($C000)
|_____|
| BANK3 16K SPEECH DATA    (233) |
|                                | 32768 ($8000)
|_____|
| BANK2 16K SPEECH DATA    (229) |
|                                | 16384 ($4000)
|_____|
| BANK1 15.5K SPEECH DATA  (225) |
|- - - - - - - - - - - - - - -   |-331 ($14B)
|    SPEECH FILE POINTERS         |
|_____| 0
```

ZERO PAGE: All versions use locations 207 through 211 and these locations are not available to the user. Locations 214 through 255 (except 251) are 'shared' with BASIC and available for user application, but any data stored here will be 'wiped-out' when BASIC or the MASTER program accesses them. Locations 203 through 206 are not used by either BASIC or the MASTER program, and are 'free' for user application.

STACK: Location 256 is used to store the MASTER program version identification number. The value in this location, after program initialization, will be:

                    1 for VM.800 version
                    2 for VM.XL   version
                    3 for VM.XE   version

PAGE 6: The upper half of 'page 6' (128 bytes) is used to store a 'jump table' for the command routines. The table is only necessary for operation of the wedge. If you absolutely need to use this area of RAM, then you can bypass the table with direct jumps to the command routines via the USR(X) statement. Careful examination of this area will reveal where the command routines are located in the MASTER program. Versions VM.XL and VM.XE use an additional 83 bytes (from 1537 to 1619) to store the ROM switching routine that allows access to the RAM behind the BASIC and O.S. ROMs.

MEMORY FROM 7420 TO 40960: The 1K area of memory, from 39936 to 40960, is ideal for PM graphics and character set storage (provided that the 'BARGRAPH' program is not used). Other details about these areas of memory have been given elsewhere in the manual. See, particularly, APPENDIX I.

BEHIND BASIC and O.S. ROM: On the Atari 800XL and 130XE, this area (40960 to 65535) is available as free RAM via 'ROM switching' techniques. The VM.XL and VM.XE MASTER programs are located 'behind' the BASIC ROM and accessed with the wedged commands via the routines on PAGE 6.

The complete Atari character set has been duplicated in the RAM from 57344 to 58368. This was necessary to preserve the screen display during O.S. ROM switching. If you want to use a custom character set with the VM.XL and VM.XE MASTER programs, then you will have to copy it to this area of memory as well. The RAM area from 55296 to 57344 is free for user application.

```
********************************
         APPENDIX V
  Notes On The Voice Master Wedge
********************************
```

The BASIC wedge used in the Voice Master program converts a command word like LEARN or SPEAK into a USR(X) statement. For instance, when the command LEARN 5 is entered, the wedge will return 'LEARN=USR(1670,5)'. BASIC interprets 'LEARN' as a variable which has a value equal to the decimal address (LEARN=1670) of the USR(X) routine. The addresses 1670 and 1670+1 on the VM.800 version of the MASTER program (1672 and 1672+1 on XL and XE versions) contain the low and high bytes of the address where the actual LEARN routine begins. If you want to see how the wedge works, just type a few BASIC program lines containing Voice Master commands and then LIST the program.

Because of the way that the wedge checks for correct syntax of wedged Voice Master commands, there are some minor restrictions that must be observed. When entering Voice Master commands in BASIC program lines:

1. The Voice Master command must immediately follow the colon in multiple statement program lines. For example:

    10 FOR X=1 TO 10:LEARN X:NEXT X

This is acceptable to the wedge. However:

    10 FOR X=1 to 10: LEARN X:NEXT X

is not acceptable because of the space after the colon and the wedge will not convert the LEARN X command to the USR(X) form.

2. To use a Voice Master command with an 'IF THEN' statement on the same program line, the command must be preceded with an ':' and a '?' must follow the 'THEN'. Enter the line in this manner:

    10 IF X=10 THEN ?:SPEAK X

Then LIST the line and the SPEAK X will be converted to the USR(X) form:

    10 IF X=10 THEN ?:SPEAK=USR(1676,X)

Now the '?' and the ':' can be edited out and the line, when listed again, will appear as:

    10 IF X=10 THEN SPEAK=USR(1676,X)

which is acceptable to BASIC. (Note: the question mark, '?', is the abbreviated form of the PRINT statement in Atari BASIC.)

The alternative method of using a Voice Master command with an 'IF THEN' statement is to put the Voice Master command on a separate program line and use the 'IF THEN' statement as a pointer to that line. For example:

    10 IF X=10 THEN 40
    20 REM These are other
    30 REM program lines.
    40 SPEAK X

3. Another 'peculiarity' is that: the wedge will always convert a Voice Master command that immediately follows a colon to the USR(X) form. With this program line:

    10 REM This is an example of a Voice Master command:SPEAK 1

the wedge would convert the 'SPEAK 1' to the USR(X) form and the line, when

listed to the screen, would appear as:
        10 REM This is an example of a Voice Master command:SPEAK=USR(1676,1)

This means that, if you do not want the wedge to convert the Voice Master command to the USR(X) form, then you must be careful not to precede the command with a colon in a BASIC REM, PRINT or DATA statement.

    4. Unlike Atari BASIC commands, a device must be specified as part of the filename when entering the SSAVE, SLOAD, TSAVE and TLOAD commands. For example, the statement:
        10 SLOAD FN$
will not be accepted by the wedge.
However,
        10 SLOAD"D:FN$"
is acceptable and when listed will appear as:
        10 SLOAD=USR(1766,ADR("D:FN$"))
Once the line is in the above form, the filename and quotes can be edited out so that the line will look like this:
        10 SLOAD=USR(1766,ADR(FN$))
which is equivalent to:
        10 SLOAD FN$

    If you keep in mind the above limitations when writing your BASIC programs, then you will have no problem with the wedged commands.

****
APPENDIX VI
Phonetic Alphabet & Numbers
****

Phonetic alphabet

Alpha	Bravo	Charlie	Delta	Echo	Foxtrot	Golf	Hotel
India	Juliette	Kilo	Lima	Mike	November	Papa	Quebec
Romeo	Sierra	Tango	Uniform	Victor	Whiskey	X-ray	Yankee
Zulu							

Airman's numbers

Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Niner

Telephone operator's numbers

Oh	One	Two	Thuh-ree	Fow-wer	Fie-yuv	Six	Seven
Eight	Nine (or Nie-yun)						

COVOX INC.

675-D Conger Street
Eugene, OR 97402
503/342-1271
Telex 706017 (Av Alarm UD)
Fax 503/342-1283